

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
ESCUELA DE POSGRADO



**IMPLEMENTACIÓN DE UN ALGORITMO DE GRILLA MÓVIL PARA LA
SIMULACION DE REACCIONES AUTOCATALÍTICAS**

Tesis para optar el grado de Magíster en Informática que presenta
CESAR HILARIO MANCILLA AGUILAR

Dirigido por
DR. CESAR BELTRAN CASTAÑON
DR. DESIDERIO VASQUEZ RODRIGUEZ

San Miguel, 2016

INDICE

1. Resumen.....	3
2. Introducción	5
3. Presentación del problema.....	13
3.1. Definición del problema	13
3.2. Objetivo general.....	14
3.3. Objetivos específicos.....	14
3.4. Resultados esperados	15
3.5. Justificación	15
3.6. Límites del proyecto.....	16
4. Marco conceptual	16
4.1. Aproximación de la ecuación FK.....	16
4.2. Valores conocidos de las velocidades de los frentes de onda	18
5. Revisión del estado del arte	19
6. Implementación de la grilla móvil por diferencias finitas.....	20
6.1. Condiciones iniciales y de frontera.....	21
6.2. Magnitud de las dimensiones espacio-temporales a utilizar	21
6.3. Consideraciones de estabilidad y convergencia.....	24
6.4. Mecanismo de la grilla móvil	25
6.5. Memoria cache L1-L2-L3 y selección del tamaño del bloque para la grilla móvil.....	27
6.6. Procedimientos generales asociados al uso de la grilla móvil	29
6.7. Implementación de la grilla en Fortran	30
6.8. Cálculo de la velocidad del frente de onda	32
7. Validación de resultados del algoritmo – orden de reacción 2 y 3	33
8. Pronóstico de resultados del algoritmo – orden de reacción 4 y 5.....	35
9. Conclusiones	41
10. Trabajos Futuros.....	43
Anexo 01: Velocidad del frente de onda - reacción cúbica, $U = 0$	44
Anexo 02: Demostración que $f_n(c) > f_{n+1}(c)$, cuando $0 < c < 1$	45
11. Referencias.....	46

1. Resumen

En este trabajo se presenta el diseño e implementación de una grilla (ventana multidimensional) móvil para aproximar numéricamente la solución de la ecuación de Fisher-Kolmogorov (FK), aplicada al modelamiento de frentes de reacciones autocatalíticas de reacción-difusión-advección. Nos apoyaremos en la perspectiva de la físico-química, el análisis numérico y su implementación en computador. Explicaremos concretamente en qué consisten las reacciones autocatalíticas y la importancia del estudio de los frentes de onda en dichas reacciones. Precisamos los fenómenos de reacción-difusión, mencionando los principales valores conocidos de velocidad de los frentes de onda, cuando no existe flujo externo. Mencionamos la dinámica de los sistemas de reacción-difusión-advección, con velocidades externas pequeñas (número de Peclet pequeño ($Pe \ll 1$)), haciendo énfasis en el fenómeno de la dispersión hidrodinámica de Taylor y algunas aproximaciones asociadas.

En cuanto al análisis numérico, utilizamos el método de diferencias finitas para aproximar las ecuaciones diferenciales mencionadas anteriormente, convirtiéndolas en relaciones algebraicas, utilizando el método Forward-Time Centered-Space (FTCS), mencionando los criterios de convergencia y estabilidad necesarios. Con ello, formulamos y sustentamos un algoritmo de grilla móvil, implementándolo en Fortran.

Con el algoritmo implementado en Fortran, simulamos los frentes de onda, sin advección y con advección, comparando los valores teóricos de velocidades de frentes de onda, con los valores simulados. Al obtener una similitud razonable entre los valores de velocidad de los frentes de onda, para reacciones de orden dos (cuadráticas) y tres (cúbicas), procedimos a realizar predicciones sobre las velocidades de los frentes para reacciones de orden superior, cuatro y cinco. Durante las simulaciones por computadora, hemos conseguido mejorar el procesamiento de datos utilizando procesamiento paralelo y operaciones de bloques de memoria, adaptando nuestras estructuras de datos a la jerarquía de memoria cache del CPU utilizado.

El algoritmo de grilla móvil implementado, con su estrategia de procesamiento y adaptación de estructuras de datos, puede ser adaptado fácilmente para estudiar otros fenómenos que admiten un esquema explícito para ser simulados.

Abstract

In this work we show the design and implementation of a moving grid to solve numerically nonlinear Fisher Kolmogorov's equation (FK), to model autocatalytic reaction-diffusion-advection fronts. We explain briefly what autocatalytic reactions are and the importance of studying their fronts. We focus on reaction-diffusion, mentioning the main known values for velocity fronts when there is not external flux. We talk about reaction-diffusion-advection systems, with small external velocity flux (Peclet's number $Pe \ll 1$), pointing out Taylor's hydrodynamic dispersion and some related approximations.

Using Numerical Analysis, we choose finite difference method to approximate our partial differential equations, turning them into algebraic relationships. We use FTCS explicit method, recalling suitable stability and convergence criteria. We enunciate a moving mesh algorithm, showing its implementation through Fortran 95.

To check our work, we simulate autocatalytic reaction fronts without and with external flux, comparing theoretical front velocity values with simulated ones. Having got a reasonable agreement for quadratic and cubic reactions, we proceed to simulate velocity fronts for autocatalytic reactions with four and five orders. Along simulation's process we have got faster results using parallel processing and memory burst operations, adapting our data structures to available CPU memory caching.

The implemented moving grid algorithm, with its processing strategy and data structure features, can be easily adapted to study other phenomena that admit a specific explicit scheme to simulate them.

Palabras clave

Velocidad Frentes de Onda, Reacciones autocatalíticas, Ecuación de Fisher Kolmogorov (FK), Grilla Móvil, FTCS, GNU Fortran

Keywords

Autocatalytic reaction front velocities, Fisher-Kolmogorov equation (FK), moving grid, FTCS, GNU Fortran.

2. Introducción

El estudio de las reacciones químicas autocatalíticas y sus respectivos frentes de onda es un problema relevante del ámbito de la Físico-Química que se estudia desde inicios del siglo XX [1]. La difusión cumple una tarea importante en las reacciones en mezclas que no son agitadas. El esparcimiento de la sustancia autocatalítica por la difusión inicia la autocatálisis en regiones vecinas: la reacción autocatalítica se acopla con la difusión molecular para dar una forma de onda constante. En este tipo de reacción, se tienen dos reactantes, uno de los cuales cumple el rol de auto catalizador de la reacción, es decir, es también producto de la reacción, con diferente orden de reacción, como por ejemplo la sustancia B en las reacciones $A + B \rightarrow 2B$ (*orden 2*) y $A + 2B \rightarrow 3B$ (*orden 3*), reacciones en las cuales la sustancia B ayuda a la formación de sí misma.

De acuerdo a Masere et. al [32], el estudio de la velocidad de los frentes de onda ayuda a entender de forma simple los mecanismos de la reacción y difusión asociados. Además, propone que la reacción química se produce en la forma de una frontera delgada que separa los dos fluidos diferentes implicados en la auto catálisis: esto es posible, pues el espesor de la región donde se produce la reacción es pequeña comparada con las otras dimensiones presentes en la autocatálisis. En otras palabras, una cantidad pequeña de la sustancia B se añade a una mezcla que inicialmente contiene la sustancia A para comenzar la reacción: los reactantes se convierten en producto en la forma de un frente que se propaga, donde la reacción se produce en la frontera que divide los reactantes y el producto.

Adicionalmente, la interacción entre la reacción, la difusión molecular y un flujo externo genera diversos mecanismos para formación de patrones (frentes de onda) [33]. En la década de 1950 el matemático inglés Alan Turing [34] predijo la existencia de patrones químicos en las reacciones de reacción difusión – con difusividades distintas para cada compuesto, patrones que se observaron experimentalmente cuarenta años después, al interior de gels de poliacrilamida [35].

También es importante conocer qué consecuencias tiene aplicar un flujo cortante a la difusión y reacción. Es preciso mencionar que un flujo cortante es la aplicación de un

flujo externo que varía de acuerdo a la posición en la cual se presenta, flujo que puede tener una dirección paralela o perpendicular al movimiento de la reacción autocatalítica.

Vásquez [33] menciona que al aplicar un flujo cortante a la reacción química del yodato ácido arsenioso se obtienen resultados diferentes en la forma y en la velocidad de los frentes de onda, dependiendo si el frente de onda se propaga en el mismo sentido o en sentido contrario al flujo externo [36], [37].

Tenemos como ejemplos concretos de reacciones autocatalíticas las del yodato ácido arsenioso, la oxidación por bromato del ácido arsenioso, la oxidación por permanganato del ácido oxálico, etc. [1], [4], [10].

Los frentes de reacción químicos tienen un comportamiento similar con otros fenómenos, tales como ondas de propagación de impulsos nerviosos, actividad eléctrica muscular, etc. La ecuación diferencial que nos ayuda a modelar los frentes de onda de reacciones autocatalíticas es una ecuación formulada por el científico Fisher en 1933, que buscaba modelar la propagación de un gen en un ser vivo, una ecuación de reacción-difusión, que también fue planteada de forma independiente por el científico ruso Kolmogorov y sus colegas, razón por la cual esta ecuación se conoce como ecuación de Fisher-Kolmogorov-Petrovskii-Piskounov (FKPP) [4], que es una ecuación diferencial parcial no lineal cuadrática (orden 2):

$$\frac{\partial p}{\partial t} = D \frac{\partial^2 p}{\partial x^2} + kp(1 - p)$$

Donde la variable p es la frecuencia de un gen mutante mientras este gen se disemina dentro de una población como un frente de propagación.

La ecuación FKPP admite soluciones cuando se tiene soporte compacto, que es cuando la sustancia autocatalítica se añade en regiones específicas (pequeñas y/o acotadas) sobre la sustancia reactante inicial. Inclusive en este caso, la onda que se forma depende de la velocidad v con la cual se traslada el frente de onda. A este mecanismo se le conoce como selección de velocidad (frente de onda) [1].

La ecuación FKPP admite una solución cerrada conocida [29], [1], para $v = \left(\frac{\pm 5}{\sqrt{6}}\right)$. No se

conoce una solución general conocida para otras velocidades; sólo se conoce que la velocidad v de los frentes de onda asociados a la ecuación FKPP en general, cumple con la condición $v \geq 2\sqrt{kD}$ [1], [4].

En la presente tesis tomamos como base una investigación de Leconte et al. [10], donde se usa una forma algo diferente de la ecuación FKPP, denominada Ecuación de Fisher Kolmogorov (FK):

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + c^2(1 - c), \dots \dots (1)$$

Puede observarse que el exponente cuadrático está fuera del paréntesis **(1-c)**. La ecuación (1) representa una reacción autocatalítica cúbica. A diferencia de la ecuación FKPP, la ecuación FK si tiene solución analítica conocida. En el Anexo 01 se describe esta solución.

A diferencia de la ecuación FKPP, con su no linealidad cuadrática y caracterizada por el trabajo trascendental de Fisher (1933), no se dispone a la fecha de una caracterización similar para la ecuación FK, con su no linealidad cúbica.

Es importante y pertinente el estudio de las reacciones autocatalíticas de orden cúbico, pues hasta hace pocos años (década de los 90s), estas reacciones no habían sido estudiadas de forma tan profusa como las ecuaciones no lineales cuadráticas. Además, entre ambos tipos de no linealidad se presentan diferencias: por ejemplo, para el caso de ondas planas (2+1 dimensiones) se tienen inestabilidades laterales en las reacciones cúbicas, lo cual no se presenta en las reacciones autocatalíticas cuadráticas [1].

Se tienen como ejemplo de reacciones cúbicas las llamas frías en la combustión del CS_2 [1], y la reacción del yodato ácido arsenioso [10].

Volviendo a la ecuación (1) – en forma canónica, esta ecuación está referida al yodato ácido arsenioso, que tiene una reacción autocatalítica de orden 3 (cúbica). En su forma no canónica, la ecuación (1) se convierte en:

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + \alpha c^2(1 - c), \dots \dots \dots (2)$$

En la cual c representa la concentración normalizada del producto de la reacción, el parámetro D se refiera a la difusividad molecular y el parámetro α es una medida de la

relevancia de la reacción autocatalítica cúbica. La expresión para la velocidad del frente de reacción autocatalítica y del ancho de dicho frente, son [10]:

$$V_x = \sqrt{\frac{\alpha D}{2}} \text{ (velocidad)}, \quad l_x = \sqrt{\frac{2D}{\alpha}} \text{ (ancho del frente)} \dots\dots\dots (3)$$

Para profundizar con el análisis de la reacción cúbica, es importante aumentar el efecto de otro fenómeno, como la advección, por lo cual necesitamos pasar de una ecuación diferencial en 1+1 dimensiones (una dimensión espacial y una temporal) hacia una ecuación en 2+1 dimensiones (dos dimensiones espaciales y una temporal).

Leconte et al. [10], discuten el efecto de un flujo externo transversal, en 2+1 dimensiones, la cual es modelada mediante la ecuación de reacción-difusión-advección (ADR):

$$\frac{\partial c}{\partial t} + u(y) \frac{\partial c}{\partial x} = D \left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} \right) + \alpha c^2 (1 - c) \dots\dots\dots (4)$$

Donde c representa la concentración normalizada de un frente de reacción que se desplaza en el eje X , con una velocidad transversal $u(y)$ utilizada por Chatwin [12]:

$$u(y) = U_{max} \left(1 - \frac{4y^2}{b^2} \right), b: \text{ancho de la celda HS} \left(-\frac{b}{2} \leq y \leq \frac{b}{2} \right), \text{ con } \bar{U} = \frac{2U_{max}}{3} \dots\dots (5),$$

donde celda HS: celda Hele Shaw.

De acuerdo a lo expresado por Leconte et al. [4], cuando tenemos un valor de α nulo (o mucho menor que la difusividad molecular D), estamos ante una reacción de difusión-advección, que a nivel macroscópico resulta en un fenómeno de dispersión hidrodinámica, denominado régimen de Taylor [30].

El científico G.I. Taylor identificó una expresión matemática para cuando una pequeña cantidad de disolución se inyecta en un tubo capilar, por el cual está fluyendo una disolución con los mismos componentes que la primera, pero a distinta concentración: el pulso de disolución inyectada se mueve a lo largo del tubo, como si fuera una columna simétrica en crecimiento lento [31].

Retomando nuestra ecuación ADR para cuando α es nulo (o insignificante con la difusividad molecular), Leconte [4] menciona que se tiene un aumento neto en la difusividad molecular, expresado mediante una **difusión efectiva**:

$$D_{eff} = D + D_T$$

Donde D_T es el aporte del régimen de Taylor a la difusividad. Leconte et al. [10] menciona que en el espacio (gap) entre 2 placas se puede establecer la siguiente relación:

$$\frac{D_{eff}}{D} = 1 + \frac{1}{210} \left(\frac{\bar{U}b}{D} \right)^2 = 1 + \frac{2}{945} \left(\frac{b}{l_x} \right)^2 \left(\frac{U_{max}}{V_x} \right)^2 \dots \dots \dots (6)$$

En la expresión (6) podemos identificar que, $D_T = \frac{\bar{U}^2 b^2}{210D}$ Además, para poder entender mejor el contexto físico del fenómeno, si se toman en cuenta las magnitudes adimensionales $\Phi = \frac{b}{l_x}$ (*ancho reducido*) y $|\epsilon| = \frac{U_{max}}{V_x}$ (*velocidad reducida*), luego de algo de álgebra, se puede establecer la siguiente relación:

$$\frac{V_f - \bar{U}}{V_x} = \sqrt{\frac{D_{eff}}{D}} = \sqrt{1 + \frac{2}{945} (\epsilon\Phi)^2} \dots \dots \dots (7)$$

Lo importante de la relación (7) es que para reacciones autocatalíticas de orden tres, nos permite hacer una predicción teórica de la velocidad de un frente de onda V_f en 2+1 dimensiones, a partir de la velocidad V_x en 1+1 dimensiones, sobre la base del promedio de la velocidad transversal promedio \bar{U} y la proporción relativa entre el ancho transversal de la celda Hele Shaw (**b**) y al ancho del frente de onda l_x .

Habiendo hecho una pequeña introducción a la parte física del presente trabajo, es preciso visualizar la parte matemática.

Hasta ahora hemos presentado de forma general las ecuaciones diferenciales relacionadas a las reacciones autocatalíticas de orden dos y tres (FKPP y FK). Sin embargo, a la fecha no se han hecho mayores trabajos sobre reacciones autocatalíticas de orden superior (mayores a tres), que puedan ser modeladas con la ecuación diferencial parcial de la forma:

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + \alpha c^n (1 - c), \text{ con } n \geq 3 \dots \dots \dots (8)$$

En la cual el orden de la reacción autocatalítica es "**n+1**". Por similitud de la ecuación (8) con la FK, llamaremos a la ecuación (8) ecuación FK generalizada, o **FK(n+1)**.

A la fecha la ecuación FK(n+1) no tiene solución analítica cerrada para $n > 3$ (orden de la reacción autocatalítica cuatro y superior). Ni siquiera se tiene una expresión para la velocidad del frente de onda asociado, ni existen caracterizaciones o estudios sobre

cotas superiores o inferiores para las velocidades de los antedichos frentes de onda.

Haciendo un resumen de lo mencionado sobre las ecuaciones FKPP, FK y FK(n+1) consideradas hasta ahora, podemos decir:

Tabla 1. Comparación entre las ecuaciones FK(n+1), FK y FKPP

Ecuación	Orden de reacción	Solución Analítica Conocida [†]	Expresión para la Velocidad	Límites (cotas) para velocidades	Velocidad Base (1+1 dimensiones)
FK(n+1)	Cuatro y superior	desconocida	desconocida	desconocida	desconocida
FK	Tres	SI	SI	SI	$v = \sqrt{kD}/\sqrt{2}$
FKPP	Dos	NO*	Selección**	SI	Selección**

Fuente: Elaboración propia

(†) : ecuación del frente de onda, en forma de onda viajera

(*) : Sólo para casos específicos (velocidad $v = 5\sqrt{kD}/\sqrt{6}$ y soporte compacto)

(**): Selección de velocidades, y soporte compacto, con velocidad $v \geq 2\sqrt{kD}$

Sobre la base de la tabla 1, visualizamos una oportunidad y/o necesidad de utilizar el análisis numérico para todas las situaciones en las cuales no se tiene expresiones analíticas generales. Adicionalmente, las entradas donde se tienen valores o expresiones conocidas, nos sirven para validar nuestros métodos numéricos, buscando que los resultados obtenidos por simulaciones coincidan con los antedichos valores.

En otras palabras, una de las estrategias para la presente tesis ha sido usar los valores generales conocidos asociados a la ecuación FK, y el caso particular conocido de la ecuación FKPP, como referentes para validar los métodos numéricos o algoritmos que utilizamos y/o implementamos en computador.

Por lo expuesto, trabajamos en la presente tesis con la siguiente **estrategia de fases**:

Fase 1: Replicación de valores conocidos de velocidades base:

- 1.1. Réplica de la velocidad base de los frentes de onda, para la ecuación FK (reacciones autocatalíticas cúbicas, en 1+1 dimensiones).
- 1.2. Réplica de la velocidad especial del frente de onda para la ecuación FKPP, con

soporte concreto. (reacción cuadrática, en 1+1 dimensiones)

Fase 2: Estudio del comportamiento de las velocidades finales v_f de los frentes de onda, cuando se tiene un flujo externo transversal, usando la ecuación ADR (que incorpora una velocidad vertical a la dirección de movimiento del frente de onda), considerando como fijo el ancho transversal de la celda Hele Shaw ($b \text{ fijo} \sim \Phi = \frac{b}{l_x} \text{ fijo}$), con variación lenta de la velocidad reducida ϵ , como medida de variación de la velocidad promedio transversal \bar{U} , considerando para cada valor en particular de ϵ las siguientes actividades:

- 2.1. Cálculo de la velocidad final teórica, usando la ecuación (7)
- 2.2. Aproximación por métodos numéricos de la velocidad final en 1+1 dimensiones, usando la ecuación FK y difusión efectiva, con $D_{eff} = D + D_T$ y $D_T = \frac{\bar{U}^2 b^2}{210D}$
- 2.3. Aproximación por métodos numéricos de la velocidad final en 2+1 dimensiones, usando la ecuación ADR (7).

Una vez que las fases anteriores (1 y 2) se completen conforme lo esperado, quedaría validada nuestra aproximación numérica. A continuación, podemos pasar a las siguientes fases, las de **pronósticos**:

Fase 3: Pronóstico de las velocidades base para reacciones autocatalíticas de orden superior, trabajando con las ecuaciones FK(4) y FK(5), con lo cual obtenemos los valores simulados para las velocidades (base) de los frentes de onda en 1+1 dimensiones, para reacciones autocatalíticas de orden 4 y 5, a saber, v_{4s} y v_{5s}

Fase 4: Aproximación de las velocidades finales de los frentes de onda v_f y estudio de la validez de la aproximación de Taylor, cuando se tiene flujo transversal en reacciones autocatalíticas de orden 4 y 5, usando las ecuaciones FK(4) y FK(5)

Haciendo variar lentamente la velocidad reducida ϵ , se realizan las siguientes actividades:

- 4.1. Aproximación de la velocidad final de los frentes de onda, usando las ecuaciones

FK(4) y FK(5) en 1+1 dimensiones, tomando como referencia la fórmula de la difusión efectiva (5), a saber $D_{eff} = D + D_T$ y $D_T = \frac{\bar{U}^2 b^2}{210D}$

4.2. Cálculo de la velocidad final de los frentes de onda, usando las ecuaciones FK(4) y FK(5) en 2+1 dimensiones, sin usar difusión efectiva.

Debemos recordar que para el caso de las reacciones autocatalíticas de orden 4 y 5, no se tienen establecidas expresiones teóricas para la velocidad final de los frentes, por lo cual nos vemos en la necesidad de utilizar los valores aproximados v_{4s} y v_{5s} .

Sabiendo las velocidades que deseamos calcular, revisamos métodos numéricos en las referencias [5], [7], [8] y [9], para aproximar la solución de nuestras ecuaciones diferenciales parciales objetivo {FK(n+1), FK y FKPP}: diferencias finitas, diferencias de volumen y elementos finitos, entre otros. Seleccionamos [5] el método de aproximación FTCS (forward time centered space, es decir, Diferencias Centradas hacia adelante – **Explícito**), con el cual convertimos las ecuaciones diferenciales en relaciones algebraicas.

Seleccionado el método de aproximación, trabajamos la implementación del algoritmo en computadora.

De modo general, podemos mencionar que el uso de métodos numéricos ha evolucionado sustancialmente con el uso de computadoras. Los algoritmos se utilizan y refinan mediante el uso de lenguajes de programación, tales como Fortran, C, C++, Ada, etc., siendo los más utilizados Fortran y C/C++. Adicionalmente se disponen de Librerías especializadas como BLAS, LAPACK, IMSL, etc., diseñadas para facilitar el desarrollo de programas que resuelvan numéricamente diferentes problemas de análisis numérico, entre ellos ecuaciones diferenciales no lineales.

Por otro lado, también se dispone de programas de cálculo o análisis simbólico, tales como MatLab®, Mathematica® y Maple®, que han facilitado y simplificado el uso y validación de nuevos métodos analíticos para resolver ecuaciones no lineales en derivadas parciales. Nosotros utilizaremos estas herramientas posteriormente en el presente trabajo.

Durante el proceso de análisis seguido para adaptar el método FTCS a nuestras necesidades, considerando lo mencionado por U. Ebert and W. van Saarloos [14], en el sentido que las velocidades de los frentes de onda tienden a converger para valores del tiempo suficientemente grandes, se hizo patente que si deseábamos utilizar una estructura de datos estática, con dimensiones fijas, esto requería a contar con una capacidad de memoria muy superior a la memoria de la mayoría de plataformas computacionales disponibles.

Por lo expuesto en el párrafo anterior, dado que el tiempo era la dimensión más demandante, se nos ocurrió hacer una especie de modularización basada en el tiempo, lo cual nos llevó a la idea de trabajar con una grilla móvil, donde la estructura de datos a usar tenía tamaño fijo para las dimensiones espaciales y, para el caso de la dimensión temporal, se trabajaba en “bloques de tiempo”, que barrían toda la dimensión temporal mucho mayor. Esto nos condujo a la idea de una grilla móvil, cuya implementación mostramos en el presente trabajo.

El algoritmo y la estructura de datos fueron implementados con GNU Fortran, dado que es un compilador confiable de Fortran, sin costo de licenciamiento.

A nivel de programación, en el presente trabajo se han seguido recomendaciones varias para optimizar el desempeño del ejecutable generado en Fortran, e inclusive se exploró el uso de procesamiento paralelo.

Sobre la base de los resultados obtenidos en las simulaciones, finalizamos la tesis, presentando conclusiones y sugiriendo investigaciones ulteriores, sobre la base de la lógica del algoritmo implementado.

3. Presentación del problema

3.1. Definición del problema

Acorde a la tabla 1, la Ecuación de Fisher Kolmogorov generalizada $FK(n+1)$

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + \alpha c^n (1 - c), \text{ con } n \geq 3$$

no tiene solución analítica cerrada. Adicionalmente, no se disponen de expresiones

analíticas ni cotas para las velocidades de los frentes de onda de las reacciones autocatalíticas modeladas por esta ecuación diferencial parcial no lineal.

3.2. Objetivo general

Proponer e implementar en computadora un algoritmo que nos ayude a estudiar la Ecuación Generalizada de Fisher Kolmogorov $FK(n+1)$.

3.3. Objetivos específicos

Los objetivos específicos del presente trabajo están relacionados con lo mencionado líneas arriba en la Sección Introducción (páginas 10-11), en la cual se enuncia una estrategia de fases, donde cada objetivo específico corresponde a una fase:

3.3.1. Replicar valores conocidos de las velocidades base de las frentes de onda para las reacciones autocatalíticas modeladas por las ecuaciones FKPP y FK (órdenes de reacción dos y tres respectivamente).

3.3.2. Estudiar el comportamiento de las velocidades finales v_f de los frentes de onda, cuando se tiene un flujo externo transversal, usando la ecuación ADR, asociada a reacciones autocatalíticas cúbicas.

3.3.3. Pronosticar las velocidades base para reacciones autocatalíticas de orden superior (cuatro y cinco), trabajando con las ecuaciones FK (4) y FK (5).

3.3.4. Aproximar las velocidades finales de los frentes de onda v_f y estudiar la validez de la aproximación de Taylor, cuando se tiene flujo transversal en reacciones autocatalíticas de orden 4 y 5, usando las ecuaciones FK(4) y FK(5)

También ha sido un objetivo de la presente tesis el optimizar el procesamiento asociado a las simulaciones, para lo cual consideramos el siguiente objetivo:

3.3.5. Minimizar el tiempo de procesamiento asociado a las simulaciones, aprovechando los recursos de cómputo disponibles, utilizando de forma implícita programación paralela.

3.4. Resultados esperados

- 3.4.1. En lo referido al objetivo específico 3.3.1., el resultado esperado es obtener (replicar) las mismas velocidades base de las frentes de onda para las reacciones autocatalíticas modeladas por las ecuaciones FKPP y FK, enunciadas en la tabla 1, dentro de los márgenes de error asociados a la aproximación por análisis numérico.
- 3.4.2. Para el objetivo específico 3.3.2., los valores esperados son los ya calculados y/o aproximados por Leconte et al. [10].
- 3.4.3. En lo referido a los valores esperados para el cumplimiento del objetivo específico 3.3.3, la condición es que las velocidades simuladas sean del mismo orden de magnitud que las velocidades encontradas para las ecuaciones FKPP y FK.
- 3.4.4. Respecto a los valores esperados para el cumplimiento del objetivo específico 3.3.4, se espera encontrar una relación razonable entre las velocidades simuladas en 1+1 dimensiones con las velocidades simuladas en 2+1 dimensiones; se busca una relación similar a la ecuación (7).

El cumplimiento del objetivo específico 3.3.5 era una condición transversal a todos los objetivos específicos, pues en caso de tenerse tiempos de procesamiento superiores a 6 horas para alguna simulación, los parámetros y supuestos de ésta eran abandonados.

3.5. Justificación

El presente trabajo busca desarrollar un algoritmo para aproximar soluciones de la Ecuación Generalizada de Fisher Kolmogorov, $FK(n+1)$, mediante el análisis numérico. La razón para hacer esto, desde el punto de vista de la física de fluidos, es la importancia que tienen en la actualidad el estudio de las reacciones autocatalíticas, y la comprensión de los mecanismos asociados.

Desde el punto de vista matemático, la ecuación $FK(n+1)$ es una ecuación diferencial parcial no lineal de la cual no se dispone de solución analítica general, ni tampoco se tiene una caracterización para la velocidad de los frentes de onda modelados por dicha ecuación diferencial parcial, usualmente una velocidad v que forma parte de la transformación $z \rightarrow kx - vt$ que describe un frente de onda viajero, con número de

onda k y velocidad v .

Por otro lado, entendemos que los esfuerzos por comprender mejor la ecuación $FK(n+1)$ servirán, tanto para aproximar su solución mediante análisis numérico, como para tener indicios de cuál podría ser el camino para obtener al menos algunas soluciones particulares de dicha ecuación diferencial parcial (EDP).

Desde el punto de vista de la informática, se busca implementar un algoritmo eficiente basado en una grilla (ventana multidimensional) móvil, buscando minimizar el tiempo de procesamiento asociado mediante el aprovechamiento de las características específicas de los recursos de cómputo disponibles, utilizando de forma implícita programación paralela.

3.6. Límites del proyecto

En el presente trabajo hemos utilizado la implementación de programación paralela OPEN MP que está disponible en el compilador GNU Fortran. El uso de OPEN MP es coherente con la arquitectura SIMD de la laptop que utilizamos para las simulaciones, Toshiba Satellite P75-A7200, con 16Gb RAM, procesador Core I7 de 4 núcleos y 8 hilos.

Respecto de las limitaciones a nivel de Análisis Numérico, se ha omitido trabajar con una grilla (ventana multidimensional) de particiones no homogéneas, de naturaleza adaptativa, especialmente en la proximidad de la perturbación, mecanismo que nos permitiría obtener aproximaciones de mayor nivel de precisión.

Por otro lado, se ha dejado de lado el análisis más profundo de las implicancias a nivel de la Física de los resultados obtenidos para las velocidades de los frentes de onda simulados.

Para finalizar, tampoco hemos desarrollado una solución analítica cerrada de la Ecuación $FK(n+1)$, al ser la presente tesis un trabajo para optar a la Maestría en Informática..

4. Marco conceptual

4.1. Aproximación de la ecuación FK

Acorde a lo indicado por Hoffman [5], las derivadas totales y parciales pueden ser aproximadas usualmente mediante los métodos de diferencias finitas, elementos finitos

y elementos de volumen [5], [7] y [8]. Las más directas de calcular son las diferencias finitas con particiones igualmente espaciadas [5], [9]. En este sentido, para la ecuación FK en 1+1 dimensiones podemos trabajar con:

Dominio Espacial X: [0, L_x]

Longitud : L_x

Particiones : N_x

Longitud Partición: $\frac{L_x}{N_x} = \Delta x$

Expresión para X discretizado: $X_i = i\Delta x, i = 0, 1, 2, \dots, N_x$

Dominio Temporal T: [0, L_t]

Longitud : L_t

Particiones : N_t

Longitud Partición: $\frac{L_t}{N_t} = \Delta t$

Expresión para T discretizado: $T_t = t\Delta t, t = 0, 1, 2, \dots, N_t$

Sobre la base de las expresiones discretizadas, usando el Método FTCS [5], [9] (Forward Time Centered Space – Diferencias Centradas hacia adelante – **Explícito**), podemos aproximar las derivadas de una función $c(x,t) \cong c(t,x)$, mediante:

$$\frac{\partial c(t_t, x_i)}{\partial t} \cong \frac{c(t_{t+1}, x_i) - c(t_t, x_i)}{\Delta t} = \frac{c_{t+1,i} - c_{t,i}}{\Delta t}, \text{ con un error de orden } \mathcal{O}(\Delta t)$$

$$\frac{\partial^2 c(t_t, x_i)}{\partial x^2} \cong \frac{c(t_t, x_{i+1}) + c(t_t, x_{i-1}) - 2c(t_t, x_i)}{(\Delta x)^2} = \frac{c_{t,i+1} + c_{t,i-1} - 2c_{t,i}}{(\Delta x)^2}, \text{ con un error de orden } \mathcal{O}((\Delta x)^2)$$

Expresamos la función c como c(t,x), pues cuando se implementen los algoritmos en computadora, es una buena costumbre denominar las grillas de datos poniendo como dimensión primaria el tiempo. Esto permite trabajar con bloques de memoria contiguos y acelerar la velocidad de ejecución del programa.

Usando las expresiones discretizadas anteriores, reemplazando en la ecuación (2) obtenemos [5]:

$$\frac{c_{t+1,i} - c_{t,i}}{\Delta t} = D \frac{c_{t,i+1} + c_{t,i-1} - 2c_{t,i}}{(\Delta x)^2} + \alpha f(c_{t,i})$$

Simplificando, obtenemos para 1+1 dimensiones:

$$c_{t+1,i} = c_{t,i} + N_{diff1}[c_{t,i+1} + c_{t,i-1} - 2c_{t,i}] + \alpha \Delta t f(c_{t,i}), \text{ con } N_{diff1} = \frac{D\Delta t}{(\Delta x)^2} \dots\dots\dots (9)$$

Haciendo un procedimiento similar para 2+1 dimensiones:

$$c_{t+1,i,j} = c_{t,i,j} + N_{diff2}[(c_{t,i+1,j} + c_{t,i-1,j} - 2c_{t,i,j}) + (c_{t,i,j+1} + c_{t,i,j-1} - 2c_{t,i,j})] + \alpha \Delta t f(c_{t,i,j}),$$

con $N_{diff2} = \frac{D\Delta t}{(\Delta x)^2} \dots\dots\dots (10)$

Para tener convergencia en los resultados numéricos, de acuerdo a Hoffman [5], se requiere que $N_{diff1} \leq \frac{1}{2}$ y que $N_{diff2} \leq \frac{1}{4} \dots\dots (11)$

Como en nuestro trabajo queremos trabajar también con términos de convección (advección), consideremos ahora la Ec. de Fisher con término de convección [5], [10]:

En 1+1 dimensiones: $\frac{\partial c}{\partial t} + u \frac{\partial c}{\partial x} = D \frac{\partial^2 c}{\partial x^2} + \alpha f(c), \text{ con } u = cte. \dots\dots\dots (12)$

En 2+1 dimensiones: $\frac{\partial c}{\partial t} + u(y) \frac{\partial c}{\partial x} = D \left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} \right) + \alpha f(c) \dots\dots\dots (13)$

En este caso el movimiento se da en la dimensión X ; $u(y)$ varía en función de la dimensión transversal al movimiento (dimensión Y).

Aplicando el método FTCS sobre las ecuaciones (12) y (13), siguiendo las recomendaciones de Hoffman [5] sobre estabilidad, tenemos:

En 1+1 Dimensiones:

$$c_{t+1,i} = c_{t,i} + N_{diff1}[c_{t,i+1} + c_{t,i-1} - 2c_{t,i}] - N_{conv}[c_{t,i+1} - c_{t,i-1}] + \alpha \Delta t f(c_{t,i}),$$

con $N_{diff1} = \frac{D\Delta t}{(\Delta x)^2}$, $N_{conv} = \frac{u\Delta t}{2\Delta x}$, cuidándose que $N_{diff1} \leq \frac{1}{2}$ y $[N_{conv}]^2 \leq N_{diff1} \dots\dots (14)$

En 2+1 Dimensiones:

$$c_{t+1,i,j} = c_{t,i,j} + N_{diff2}[(c_{t,i+1,j} + c_{t,i-1,j} - 2c_{t,i,j}) + (c_{t,i,j+1} + c_{t,i,j-1} - 2c_{t,i,j})]$$

$$- N_{conv}[c_{t,i+1,j} - c_{t,i-1,j}] + \alpha \Delta t f(c_{t,i,j}),$$

con $N_{diff2} = \frac{D\Delta t}{(\Delta x)^2}$, $N_{conv} = \frac{u(y)\Delta t}{2\Delta x}$, cumpliendo que $N_{diff2} \leq \frac{1}{4}$ y $\max_{y \in [0, L_y]} |N_{conv}| \leq \sqrt{N_{diff2}} \dots\dots (15)$

Enfatizamos que para cada ecuación se tienen condiciones iniciales y de frontera, que explicitaremos más adelante.

4.2. Valores conocidos de las velocidades de los frentes de onda

De acuerdo a lo expresado en la tabla 1, la ecuación FKPP (canónica) admite una expresión analítica para el frente de onda, cuando se tiene soporte compacto y la velocidad (seleccionada) es $\frac{5}{\sqrt{6}}$.

Cuando tenemos la ecuación FKPP en su forma no canónica, con el parámetro de difusión molecular D y el parámetro de relevancia de la reacción autocatalítica α , ambos distintos de la unidad, la expresión para la velocidad (sin advección, en 1+1 dimensiones) es

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + \alpha(1 - c), \text{ y soporte compacto; luego, velocidad } V_2 \rightarrow 2\sqrt{\alpha D} \dots (16)$$

Por otro lado, tal como se indicó en la introducción y en la tabla 1, considerando las ecuaciones (2) y (3), la expresión para la velocidad base (sin advección, en 1+1 dimensiones), para la ecuación FK es:

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + \alpha c^2(1 - c), \dots \rightarrow \text{la velocidad } V_3 = \sqrt{\alpha D} \frac{1}{\sqrt{2}} \dots (17)$$

A modo de referencia, en el **Anexo 01** mostramos el cálculo del valor analítico para la velocidad del frente de onda en la ecuación FK, usando el análisis de ondas viajeras para ecuaciones diferenciales parciales [13], mediante el método de factorización [24], [28].

5. Revisión del estado del arte

Tal como se ha mencionado en la tabla 1, la ecuación generalizada de Fisher Kolmogorov de orden “n+1”, FK(n+1), tiene solución analítica cerrada cuando n=2, y tiene una solución específica para n=1, con soporte compacto y $v = \frac{5}{\sqrt{6}}$.

A la fecha no se dispone de una solución analítica cerrada conocida para n=3, 4, etc. Sin embargo, dada la importancia de las aplicaciones en diversos campos de las ciencias de la ecuación diferencial parcial FK(n+1), a partir de la primera década del siglo XXI se han tenido varios esfuerzos por resolver o aproximar dicha ecuación, dentro de los cuales podemos mencionar principalmente dos métodos:

5.1 Métodos para obtener la solución analítica de la ecuación FK(n+1)

Para la realización de la presente tesis, se revisó los esfuerzos desarrollados para resolver la ecuación FK(3) desde la década de los 90's hasta la fecha, donde destacan los métodos:

5.1.1 Colocación (ansatz) basada en funciones trigonométricas o hiperbólicas, incluyendo la transformada Exp-Function [25].

5.1.2 Reemplazo de la función objetivo por transformación $\frac{G}{G'}$ y sus derivados [38]

5.1.3 Uso de Transformación de Backlund [38]

5.1.4 Uso de Factorización (Cornejo-Pérez y Rosú [24], [28]).

En todos los métodos enunciados, se ha observado que el uso de software de cálculo y análisis simbólico, tales como Mathematica®, Maple® o MatLab®, contribuye a realizar los cálculos con menos errores.

Sin embargo, es necesario señalar que a la fecha, los métodos anteriores no nos ayudan a resolver la ecuación FK(4) y superiores.

5.2 Métodos para aproximar la solución de la ecuación FK(n+1)

Recordando lo señalado en la introducción del presente trabajo, los métodos de aproximación de las ecuaciones FKPP y FK, y en general, la aproximación de una ecuación diferencial parcial, implica la transformación de las derivadas parciales en fórmulas basadas en diferencias, con lo cual, la expresión en derivadas (continua) se transforma en una relación de diferencias (saltos discretos).

En la actualidad se tienen diversas formas de discretizar las derivadas, siendo los esquemas clásicos los de diferencias finitas, elementos finitos y elementos de volumen, acompañados de novedosos esquemas con un nivel de precisión elevada, tales como el uso de splines y B-splines, o el mix de métodos de aproximación puntual más la aplicación de algoritmos tipo Runge Kutta.

6. Implementación de la grilla móvil por diferencias finitas

En la introducción mencionamos que para cubrir los objetivos del presente trabajo hemos escogido el método de diferencias finitas, en su variante FTCS, y hemos mostrado la discretización realizada de las ecuaciones diferenciales, quedando pendiente establecer las condiciones iniciales y de frontera para cada uno de los casos que nos interesan. Adicionalmente, mencionamos la necesidad de “modularizar” la

dimensión temporal mediante la implementación de una grilla móvil. Pasamos a exponer los detalles:

6.1. Condiciones iniciales y de frontera

De acuerdo a Hoffman [5] y Leconte et al. [4], [10], tenemos las siguientes condiciones:

Para una 1+1 dimensiones:

1. Función a aproximar: $c(x,t) \cong c(t,x) \approx c(t,x_i) \cong c_{t,i}$
2. Condiciones de frontera:
(Condiciones de Dirichlet)
 $c(t,0)=1, c(t,L_x)=0$, para todo valor de t
3. Condiciones iniciales:
 $c(0,x)=1$, si $x \in [0, \frac{L_x}{2}]$
 $c(0,x)=0$, si $x \in] \frac{L_x}{2}, L_x]$
4. No hay condiciones de Neumann (valor de derivadas)

Para 2+1 dimensiones:

1. Función a aproximar: $c(x,y,t) \cong c(t,x,y) \approx c(t,x_i,y_j) \cong c_{t,i,j}$
2. Condiciones de frontera:
(Dirichlet)
 $c(t,0,y)=1, c(t, L_x,y)=0, t>0, -\frac{b}{2} \leq y \leq \frac{b}{2}$
(Neumann)
 $\frac{\partial c}{\partial y} = 0$, para $y=0$ e $y=L_y, x \geq 0$
3. Condiciones iniciales:
 $c(0,x,y)=1$, si $x \in [0, \frac{L_x}{2}], -\frac{b}{2} \leq y \leq \frac{b}{2}$
 $c(0,x,y)=0$, si $x \in] \frac{L_x}{2}, L_x], -\frac{b}{2} \leq y \leq \frac{b}{2}$

6.2. Magnitud de las dimensiones espacio-temporales a utilizar

De acuerdo a U. Ebert y W. van Saarloos [14], las velocidades de los frentes de difusión convergen para valores del tiempo suficientemente grandes.

En el presente trabajo, hemos tomado $b = 240$ como la magnitud de la dimensión transversal, a fines de comparar los resultados con los exhibidos por Leconte et al. [10]. Dado que hemos trabajado con Fortran, cada variable (número) en doble precisión se almacena en 8 bytes [15], [16], a partir de lo cual hemos obtenido los siguientes valores para la simulación en general (para la dimensión temporal y espaciales):

6.2.1. Para una dimensión espacial y tiempo

En este caso, calculamos el total de puntos de grilla, que resulta del multiplicar el total de puntos considerados tanto para la dimensión espacial X , como para la dimensión temporal T . En ambos casos, estamos trabajando con dimensiones divididas en partes igualmente espaciadas, por lo cual el total de memoria necesaria resulta de multiplicar el total de puntos de grilla por los 8 bytes que toma almacenar cada valor en doble precisión, resultado que podemos ver en la última columna de la tabla 2, datos de consumo de memoria para 1 dimensión espacial.

Tabla 2. Datos de consumo de memoria para 1 dimensión espacial

Tipo	Largo	Tamaño de Partición (Δ)	# de Puntos	Total Ptos. en Grilla	Total en Memoria (Gb)
Temporal T	60,000	(Δt) 0.02	3'000,000	4.5×10^{10}	3.6×10^{11} Bytes
Espacial X	2,400	(Δx) 0.16	15,000		≈ 360 Gb

6.2.2. Para dos dimensiones espaciales y tiempo

Aquí, nuevamente calculamos el total de puntos de grilla, pero ahora multiplicamos el total de puntos considerados para ambas dimensiones espaciales, X e Y , además de los puntos asumidos para la dimensión temporal T . El total de memoria que necesitamos es el producto total de puntos de grilla por los 8 bytes que toma almacenar cada valor en doble precisión. Este resultado que podemos ver en la última columna de la tabla 3, datos de consumo de memoria para 2 dimensiones espaciales.

Tabla 3. Datos de consumo de memoria para 2 dimensiones espaciales

Tipo	Largo	Tamaño de	# de Puntos	Total Ptos.	Total en Memoria
------	-------	-----------	-------------	-------------	------------------

	Partición (Δ)			en Grilla	(Gb)
Temporal T	120,000	(Δt) 0.25	480,000		
Espacial X	2,400	(Δx) 2.00	1,200	2.76×10^{10}	2.212×10^{11} Bytes
Espacial Y	240	(Δy) 5.00	48		≈ 221.2 Gb



6.3. Consideraciones de estabilidad y convergencia

Tal como se mencionó en la introducción, el uso del método explícito FTCS requiere de especial cuidado para garantizar la estabilidad y convergencia de la aproximación de las ecuaciones diferenciales.

Podemos mencionar que, durante nuestras simulaciones de la ecuación FK, buscando acelerar la convergencia de los valores de las velocidades, nosotros tratamos de usar valores más pequeños de Δx , pero el tiempo de procesamiento se hizo muy largo. Esto se debió a la aparición de valores NAN (not a number [15]) para la concentración normalizada C , generada por la oscilación de los valores de C fuera del intervalo $[-1,1]$, con el consiguiente crecimiento geométrico de los valores $f(C)$. Este incidente se resolvió verificando que el par $(\Delta x, \Delta t)$ cumpliera con la condición (14).

Por otro lado, cuando tratamos que los valores de las velocidades simuladas de los frentes de onda para la ecuación FK coincidan con más cifras significativas con los valores teóricos, considerando en la simulación valores más grandes de la dimensión temporal (L_t), los valores de C para los valores cercanos a $X = L_x$ no alcanzaban armoniosamente el valor cero, decreciendo abruptamente hacia dicho valor forzados por las condiciones de frontera impuestas. Más aún, los valores de C se repetían independientemente del valor de t , lo cual nos indicaba que el frente de onda ya había llegado a $X = L_x$. Esto se resolvió tomando progresivamente valores más pequeños para L_t , hasta que los valores de C mostraban un comportamiento más armonioso.

El trabajar con 2+1 dimensiones - ecuación (13), nos propuso un reto adicional, dado que trabajar con iguales valores para Δx y Δy nos habría conducido a requerimientos de memoria mayores a los arriba indicados. La solución fue trabajar con valores diferentes de Δx y Δy , asumiendo que la constante de Difusión Molecular es la misma en ambas dimensiones espaciales, es decir, en lugar de trabajar con un solo valor N_{diff2} nos vimos obligados a trabajar con coeficientes diferentes N_{diff2x} y N_{diff2y} . De esta forma, la ecuación (10) se convirtió en:

$$c_{t+1,i,j} = c_{t,i,j} + N_{diff2x}(c_{t,i+1,j} + c_{t,i-1,j} - 2c_{t,i,j}) + N_{diff2y}(c_{t,i,j+1} + c_{t,i,j-1} - 2c_{t,i,j}) - N_{conv}[c_{t,i+1,j} - c_{t,i-1,j}] + \alpha \Delta t f(c_{t,i,j})$$

Donde $N_{diff2x} = \frac{D\Delta t}{(\Delta x)^2}$, $N_{diff2y} = \frac{D\Delta t}{(\Delta y)^2}$, $N_{conv} = \frac{u(y)\Delta t}{2\Delta x}$, parámetros que deben cumplir
 $N_{diff2x} \leq \frac{1}{4}$ y $N_{diff2y} \leq \frac{1}{4}$ y $Max_{y \in [0, L_y]} |N_{conv}| \leq \sqrt{N_{diff2x}}$ (18)

Los valores indicados en la tabla 3 cumplen las condiciones de estabilidad y convergencia dados por la relación (15).

6.4. Mecanismo de la grilla móvil

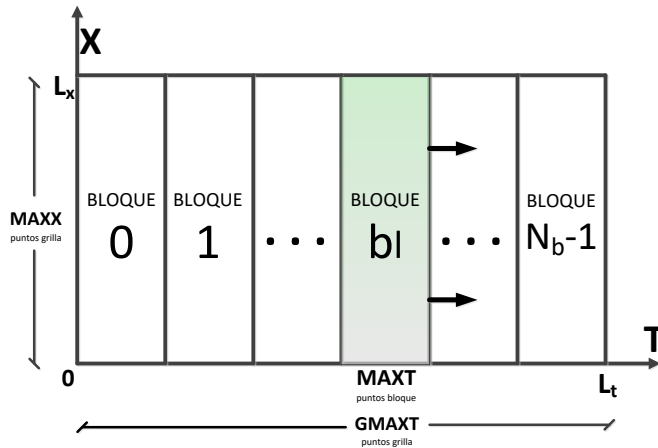
Las necesidades de memoria indicadas en las tablas 2 y 3 sobrepasan largamente lo disponible en las máquinas desktop y workstations actuales, donde un número máximo de memoria anda por 128Gb para workstations y servidores. Es más, lo usual promedio en equipos de oficina es de 8Gb, para equipos con procesador Core i7.

Podríamos argüir que para el caso de una dimensión (tabla 2), los tamaños de las particiones son muy pequeños y que podríamos trabajar con los tamaños de partición seleccionados para el trabajo con dos dimensiones espaciales (tabla 3). Sin embargo, esta selección fue hecha para tener mayor confiabilidad en los cálculos en 1 dimensión espacial. Estos valores nos ayudaron luego a comparar y validar los resultados obtenidos para dos dimensiones espaciales, mediante la relación (7).

Por otro lado, al analizar las fórmulas de aproximación y/o discretización (8) – (14), podemos percatarnos que todas tienen la forma $C_{t+1, \mathbf{r}} = f(\mathbf{r}_t)$, donde \mathbf{r} representa las coordenadas espaciales: esto es, para calcular los valores de la concentración para un instante $t+1$, necesitamos solamente los valores del instante anterior t . Por el contrario, para los algoritmos implícitos, se tiene que $C_{t+1, \mathbf{r}} = f(\mathbf{r}_t, C_{t+1, \mathbf{r}})$.

Generalizando esta dependencia uniforme $C_t \rightarrow C_{t+1}$, podemos imaginar a nuestra grilla gigante de datos, similar a un paralelepípedo de datos, barrida y/o calculada virtualmente utilizando una grilla móvil mucho más pequeña, guardando en el disco duro los datos necesarios, para cada bloque de datos que procesamos.

Por ejemplo, para simular la ecuación de Fisher en 1+1 dimensiones, ecuaciones (9) y (12), armamos nuestra grilla de datos asumiendo que la dimensión T se dividirá en N_b partes, tal como se aprecia en la figura 1.



L_x : largo dimensión X

L_t : largo dimensión T

MAXX: # Puntos grilla eje X

G_{MAXT} : # Puntos grilla eje T

MAXT: # Puntos grilla bloque

N_b : Número de bloques

$$N_b = \frac{G_{MAXT}}{MAXT}$$

Figura 1. Esquema de datos, simulación en 1+1 dimensiones

Lo que podemos ver, es que en lugar de trabajar con toda la grilla de $G_{MAXT} \times MAXX$ puntos, lo que hacemos es trabajar con bloques lógicos de $MAXT \times MAXX$ puntos. Es decir, sólo nos ocupamos de separar memoria para el bloque b_l que está siendo procesado actualmente, bloque por bloque.

Ampliando este concepto, para el caso de simular la Ecuación de Fisher en 2+1 dimensiones, ecuaciones (10) y (13), volvemos a dividir la dimensión temporal T en N_b partes, con lo cual tenemos N_b bloque de datos, cada uno con $MAXX \times MAXY \times MAXT$ puntos de grilla, lo que podemos apreciar gráficamente en la figura 2.

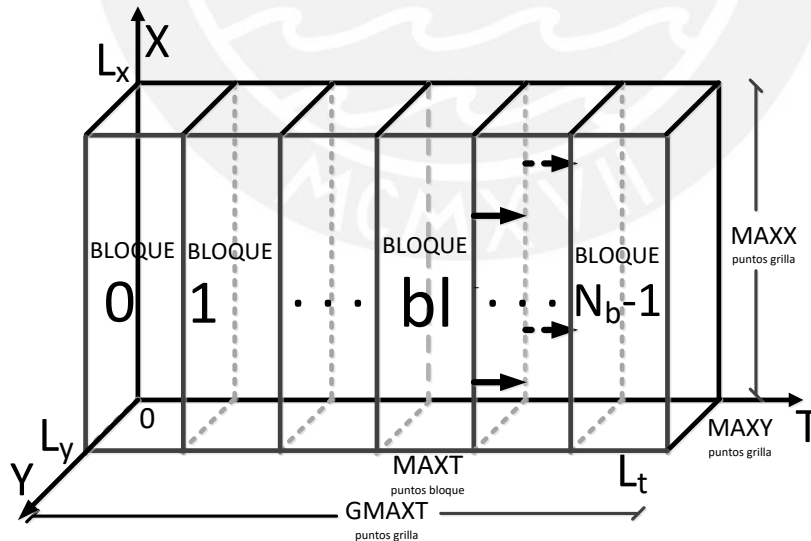


Figura 2. Esquema de datos, simulación en 2+1 dimensiones

Donde:

L_x : largo dimensión X

L_y : largo dimensión Y

L_t : largo dimensión T

MAXX: # Puntos grilla eje X

MAXY: # Puntos grilla eje Y

G_{MAXT} : # Puntos grilla eje T

MAXT: # Puntos grilla bloque

N_b : Número de bloques

$$N_b = \frac{G_{MAXT}}{MAXT}$$

Una ventaja adicional de trabajar dividiendo en bloques la dimensión temporal T , es que como los términos de error de las ecuaciones de aproximación son de orden $\mathcal{O}(\Delta t)$, lo cual nos obliga a tomar un Δt pequeño, es importante tener un mecanismo de flexibilidad y control para trabajar con un número grande de puntos GMAXT.

Es decir, mediante el uso de la grilla, en cuanto a la dimensión temporal, ya no es necesario almacenar todos los puntos de grilla del eje T (GMAXT), sino solamente nos debemos preocupar de los puntos de la grilla del Bloque que estamos procesando (MAXT). Adicionalmente, el manejo de bloques de memoria nos facilita el cálculo de las velocidades de los frentes de onda.

Pasando a otro tema, nosotros también buscamos aprovechar las capacidades de nuestra plataforma de cómputo, una laptop con procesador Core I7, aspecto que influyó de manera importante en el desarrollo del presente trabajo, tema que es materia de la siguiente sección.

6.5. Memoria cache L1-L2-L3 y selección del tamaño del bloque para la grilla móvil

En la arquitectura de microprocesadores de los equipos modernos, los diseñadores de *chips* han buscado mejorar su rendimiento utilizando conceptos adaptados del mundo *mainframe*, donde se definen distintos tipos de memoria de acuerdo al nivel en el cual interactúan con el procesador, bus interno de comunicaciones y con los periféricos.

Una de las formas más comunes de conseguir mejor rendimiento, es utilizar un **CPU cache** (memoria temporal dedicada). La CPU *cache* es una memoria rápida ubicada entre el procesador y la memoria principal (RAM); sirve para almacenar los datos e instrucciones más usados por cada núcleo del procesador [17], [18]. Sin embargo, el costo por Mb de una CPU cache es mayor que el de una RAM usual.

La mayoría de CPUs tienen diferentes memorias *cache* independientes, para almacenar tanto instrucciones como datos, donde estas caches se organizan como una jerarquía de varios niveles (L1-L2-etc). Para el presente trabajo, buscando conseguir tiempos de procesamiento aceptables, escogimos como plataforma de hardware una laptop Toshiba Satellite, modelo P75-A7200, con microprocesador Core I4 4770MQ de

4ta generación, 4 núcleos y 8 hilos, con una estructura de memorias *cache* L1 – L2 – L3 similar al microprocesador, Core I7 4900MQ, que mostramos en la figura 3:

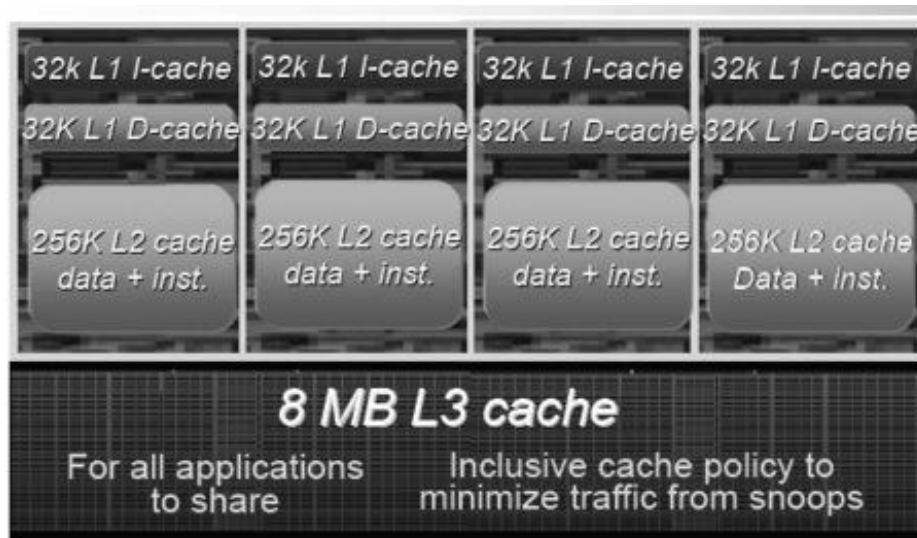


Figura 3: Estructura de Memoria *Cache*, Microprocesador Intel Core I7 4900MQ

Fuente: referencia [19]

La única diferencia entre los microprocesadores Core I7 4900MQ y 4770MQ es que el primero tiene 8Mb de *Cache* L3 en lugar de los 6Mb de *Cache* L3 que tiene el segundo.

La memoria L1, llamada también *cache primaria*, es una memoria estática integrada con el núcleo del microprocesador, memoria que almacena información recientemente accesada por el microprocesador.

El propósito de esta memoria es mejorar la velocidad de acceso a los datos cuando el CPU accede repetidas veces a la misma data. Para lograr esto, el tiempo de acceso de la memoria L1 es mucho más rápido que la memoria del sistema (RAM).

La memoria L2, llamada también *cache secundaria*, almacena información recientemente accesada. Su objetivo es reducir el tiempo de acceso cuando el mismo dato ha sido accesado anteriormente. La memoria L2 no es tan veloz como la memoria L1 pero es usualmente más grande. Todos los datos que se piden a la memoria L2 se copian en la memoria L1.

La memoria L3, es la más grande de las memorias cache: en ella se guardan datos de cualquiera de los núcleos del microprocesador, con estrategias específicas de gestión de colas.

Para nuestro microprocesador Intel Core I7 4770MQ, tenemos las siguientes capacidades de sus memorias cache (tabla 4):

Tabla 4: Memoria cache del microprocesador Intel Core I7 4770MQ

Cache	Instrucciones	Datos	Instrucciones y/o Datos
L1	4* x 32Kb	4* x 32Kb	-----
L2	-----	-----	4* x 256Kb
L3	-----	-----	6Mb

Se indica 4 x (...) cuando se distribuye la memoria por cada núcleo

Durante las simulaciones realizadas con nuestro equipo Core I7, asumimos la estrategia de ajustar el tamaño del bloque a procesar con las memorias cache L1-L2-L3 disponibles, mediante la variación del tamaño del parámetro MAXT, el cual es inversamente proporcional al número de bloques de trabajo N_b que eran recorridos por la grilla móvil.

Mediante la calibración cuidadosa del parámetro MAXT para cada simulación, se consiguieron reducciones significativas en el tiempo de procesamiento de las simulaciones: de 28 horas a 7 minutos para 1+1 dimensiones y de 75 horas a 80 minutos para 2+1 dimensiones.

6.6. Procedimientos generales asociados al uso de la grilla móvil

A nivel lógico, utilizar la grilla móvil tiene distintos aspectos que debemos cuidar, independientemente del lenguaje o entorno de programación en el cual la implementemos, tomando como base las condiciones iniciales y de frontera indicadas en el apartado (6.1), siguiendo además las recomendaciones de Hoffman [5]:

6.6.1. Pasos iniciales con primer bloque a procesar

- a. Poner a cero los valores de la grilla móvil **C** y el de todas las variables utilizadas. Esto incluye una grilla móvil **c0**, de una dimensión menor, que sirvió para almacenar los valores de interface entre los bloques, correspondientes a $t=MAXT$, que en el siguiente bloque serán los valores iniciales para $t=0$.
- b. Inicializar los valores de la grilla para $t=0$, de acuerdo a la sección 6.1

Para 0 <= Bloque <= Numero de Bloques -1, repetir

- 6.6.2. Copiar valores de grilla **c0** en grilla móvil, para $t = 0$
 - 6.6.3. Calcular el bloque corriente: **CalcBlock()**,
 - 6.6.4. Guardar en disco los datos necesarios para calcular la velocidad de los frentes, para los bloques seleccionados: **SaveBlock ()**
 - 6.6.5. Guardar en la grilla **c0** los valores de grilla **C** para $t=L_t$ (**MAXT**)
 - 6.6.6. Calcular la velocidad de los frentes y almacenar los resultados finales
- #### 6.7. Implementación de la grilla en Fortran

La presente investigación se desarrolló con el lenguaje de programación Fortran, que es el lenguaje de programación científica más ampliamente utilizado [21]. Buscando reducir costos hemos utilizado el compilador GNU Fortran, complementado con el entorno de programación CodeBlocks, ambos sin costo de licenciamiento y de libre disponibilidad en Internet. El compilador GNU Fortran se obtuvo del site www.mingw.org y el entorno de programación CodeBlocks está disponible en el site www.codeblocks.org.

Dado que el foco principal del presente trabajo es mostrar la implementación en computadora de un algoritmo, es preciso que dicha implementación sea hecha utilizando las mejores prácticas y facilidades disponibles que provean las herramientas seleccionadas. En este contexto, se revisó bibliografía sobre Fortran nivel avanzado, por lo cual optamos por seguir algunas recomendaciones señaladas por Willé [20]:

- 6.7.1. Declarar la grilla C que almacena la grilla móvil y la grilla auxiliar c0, poniendo como primer índice el tiempo. Esto es, declarar la grilla principal como C(MAXT,MAXX,MAXY) en lugar de definirla como C(MAXX,MAXY,MAXT).

Esto se probó al inicio de la programación y el ahorro de tiempo es sustancial, pues los datos se almacenan en memoria en bloques contiguos a cómo serán utilizados, lo cual favorece directamente el uso de la memoria cache (L1-L2-L3).

- 6.7.2. Utilizar siempre que sea posible la instrucción **forall**, pues se compila utilizando todos los núcleos disponibles del procesador (paralelismo), por ejemplo:

5.7.2.1. Inicialización de arreglos de memoria

5.7.2.2. Copia masiva de datos entre arreglos

- 6.7.3. Usar en las operaciones de I/O la opción de bucle **do** implícito, que se compila también en modo *burst*, en lugar de forma secuencial. Esto es crítico para almacenar los datos en disco.

- 6.7.4. Utilizar parámetros para todos los valores sensibles y/o relevantes del programa, para facilitar la depuración y mantenimiento del mismo.

Lo que también recomienda Willé [20], pero no llegó a tener impacto en la velocidad de ejecución (y por ello no se consideró en la versión final del código), fue lo siguiente:

- 6.7.5. Eliminación de la modularización del programa, es decir, hacer un solo bloque de ejecución, para no perder tiempo en llamadas a funciones y procedimientos. Esto no tuvo impacto, y más bien hacía un poco más lento el tiempo de ejecución del programa. Entendemos que se debe a que la lógica de la grilla móvil es de naturaleza intrínsecamente modular, por lo cual los tiempos “perdidos” de llamadas a procedimiento no son relevantes comparados con los tiempos de procesamiento de cada bloque. Más bien al hacer un programa lineal se pierden oportunidades de usar instrucciones en modo paralelo o *burst*.

Además de las recomendaciones de Willé [20], intentamos usar programación paralela, poniendo en paralelo las instrucciones **do** del código, utilizando Open MP, soportado

por el compilador GNU Fortran. Los resultados también fueron desalentadores, pues el programa se puso más lento. Pensamos que esto se debe a que el compilador GNU Fortran ya había paralelizado estos comandos, por lo cual al usar las directivas \$OMP, el uso y gestión de un modelo de procesamiento **fork** que emplea variables privadas y compartidas necesarias para el paralelismo, hizo que la ejecución de nuestro programa fuese más lenta.

6.8. Cálculo de la velocidad del frente de onda

Para calcular la velocidad del frente, tomamos como referencia la posición y el tiempo en el cual la concentración tiene el mismo valor, que conceptualmente es utilizar curvas de nivel: para el mismo valor de C , consideramos los valores iniciales y finales de posición y tiempo, lo cual nos da valores Δx y Δt , de lo cual obtenemos la velocidad del

$$\text{frente } V_f = \frac{\Delta x}{\Delta t}.$$

Si la magnitud de tiempo L_t no es suficiente, los valores de velocidad para distintos valores de la concentración (C) varían, desde valores de velocidad bajos, cuando $c=0.9$ hasta valores de velocidad altos, cuando $c=0.1$.

En concordancia por lo indicado por Ebert y Saarlös [14], se resolvió esta situación aumentando la magnitud y ajustando paulatinamente la partición de la dimensión temporal (L_t), llegando a los valores indicados en los apartados 6.2.1 y 6.2.2 (GMAXT 3.0E06 y 4.8E05 respectivamente). Con estos valores, las velocidades obtenidas con las curvas de nivel para $c=0.9$ y $c=0.1$ son las mismas, dentro de las escalas de velocidades consideradas.

En este sentido, para el cálculo de la velocidad del frente *sin convección*, en 1+1 dimensiones, con orden de reacción química 2 y 3, tomando $D = 0.05$ y $\alpha = 0.0004$, se obtienen los valores mostrados en la tabla 5:

Tabla 5: Simulación de velocidades de frente de onda
($U=0$), $D = 0.05$ y $\alpha = 0.0004$, 1+1 dimensiones

Orden de reacción	Ecuación	Velocidad simulada	Velocidad teórica	Diferencia (%)
Dos	FKPP	0.0091238	0.0089443	2.0068%
Tres	FK	0.0031605	0.00316228	0.0860%

En el caso de 2+1 dimensiones, nos enfocamos sólo en las reacciones de orden cúbico, considerando el efecto de la convección (ecuaciones (10) y (13)). Para esto, utilizamos la estrategia explicada en la introducción, comparando los resultados de simular en dos dimensiones espaciales versus la aproximación en una dimensión, analizando el comportamiento de la velocidad de los frentes al aumentar poco a poco el valor u_{max} de la velocidad transversal de desplazamiento del frente, vinculada a la variación lenta del parámetro adimensional $|\epsilon| = \frac{u_{max}}{V_x}$, dada por $u(y) = u_{max} \left(1 - \frac{4y^2}{b^2}\right)$, con $-\frac{b}{2} \leq y \leq \frac{b}{2}$. Los resultados los mostramos en este mismo trabajo líneas abajo.

7. Validación de resultados del algoritmo – orden de reacción 2 y 3

De acuerdo a la tabla 5, para el caso de la ecuaciones FKPP y FK en 1+1 dimensiones, la diferencia entre los valores aproximados y los teóricos conocidos (16) y (17) es menor al 0.1%, valor razonable para nuestros objetivos

Para 2+1 dimensiones, usando específicamente la ecuación FK en 2+1 dimensiones, y la aproximación (7) – difusión efectiva, con los parámetros $D = 0.05$, $\alpha = 0.0004$ y $b = 240$, variando lentamente el ratio $|\epsilon| = \frac{u_{max}}{V_3}$, obtuvimos:

Tabla 6: Comparación simulación velocidades de frente de onda
en 2+1 dimensiones (ecuación FK, orden 3)

$ \epsilon $	\bar{U}	(A)	(B)	(C)		
		V_f teórica Ecuación (7)	V_f con diff. efectiva 1+1 Dim	D_{diff} (1D) $ (A)-(B) $	V_f simulada 2+1 Dimens.	D_{diff} (2D) $ (A)-(C) $
0.00	0.00000	0.0031623	0.0031646	2.28E-06	0.0031646	2.28E-06
0.25	0.00053	0.0037371	0.0037340	3.19E-06	0.0037334	3.72E-06
0.50	0.00105	0.0044036	0.0044017	1.93E-06	0.0043956	7.97E-06
0.75	0.00158	0.0051509	0.0051447	6.16E-06	0.0050736	7.73E-05

1.00	0.00211	0.0059652	0.0059590	6.12E-06	0.0057737	1.91E-04
1.25	0.00264	0.0068327	0.0068303	2.43E-06	0.0064914	3.41E-04
1.50	0.00316	0.0077417	0.0077388	2.92E-06	0.0072150	5.27E-04
1.75	0.00369	0.0086827	0.0086780	4.69E-06	0.0079523	7.30E-04
2.00	0.00422	0.0096482	0.0096386	9.64E-06	0.0086881	9.60E-04

donde:

- (A) : Velocidad del frente, de acuerdo a expresión teórica (7)
- (B) : Velocidad *simulada* del frente, en 1 dimensión espacial, usando la difusión efectiva
- (C) : Velocidad *simulada* del frente, en 2+1 dimensiones

También se cumple que $D_{diff}(1D) \ll \alpha((\Delta x)^2)$ y $D_{diff}(2D) \ll \alpha((\Delta x)^2)$.

Con los datos de la tabla 6, para un mejor análisis, grafiquemos el efecto de la velocidad promedio transversal \bar{U} sobre las **velocidades normalizadas** $\left(\frac{v_f - \bar{U}}{v_3}\right)$, resultado que mostramos en la figura 4.

En la figura 4 observamos que las velocidades normalizadas correspondientes a la expresión teórica (7) coinciden casi perfectamente con las velocidades obtenidas mediante la simulación en 1+1 dimensiones, usando difusión efectiva.

Asimismo, en la misma figura 4, vemos que las velocidades normalizadas obtenidas mediante la ecuación FK en 2+1 dimensiones, sólo coinciden con las velocidades normalizadas producto de la expresión teórica, sólo para valores pequeños de \bar{U} .

La coincidencia entre los valores obtenidos mediante la expresión teórica y la difusión efectiva, así como la diferencia creciente que se observa con las velocidades obtenidas mediante la ecuación FK en 2+1 dimensiones, es similar a lo indicado por Leconte et. al [10].

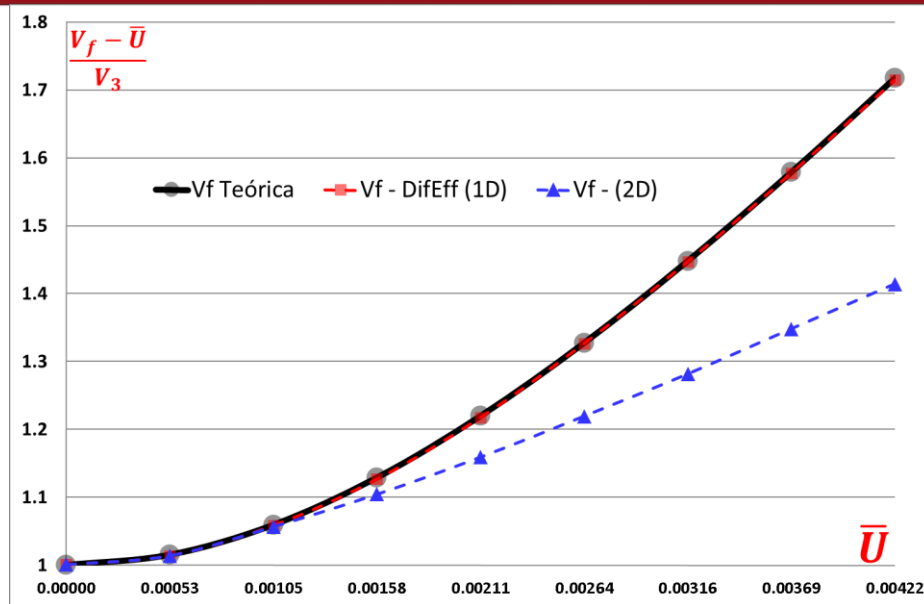


Figura 4. Velocidades normalizadas, reacción orden cúbico
 \bar{U} = velocidad promedio de flujo transversal [12, Chatwin]

Sobre la base de los resultados obtenidos, podemos asumir que la grilla móvil implementada ha simulado razonablemente las velocidades de los frentes de onda para órdenes de reacción 2 y 3. Así, conforme nuestra estrategia de trabajo mencionada en la introducción, utilizaremos nuestra grilla móvil para pronosticar:

- Las velocidades base para reacciones autocatalíticas de orden cuatro – FK(4) y de orden cinco 5 – FK(5).
- Calcular y analizar el efecto de la variación en la velocidad promedio transversal sobre las velocidades finales de los frentes de onda para reacciones autocatalíticas representadas por las ecuaciones FK(4) y FK(5).

8. Pronóstico de resultados del algoritmo – orden de reacción 4 y 5

Tal como se explicó en la Introducción del presente trabajo, Leconte et al. [10] trabajan con la ecuación FK, usando una función vinculada a la reacción autocatalítica, $f(c) = c^2 (1-c)$, donde c representa la concentración normalizada del reactante autocatalítico y el orden de la reacción es tres, que es el orden polinómico de $f(c)$.

En la introducción, por fines de simplicidad, mediante la ecuación (8), definimos la **ecuación FK generalizada, o FK(n+1)**, con orden de reacción autocatalítica “n+1”, así:

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + \alpha c^n (1 - c), \text{ con } n \geq 3, \text{ con } D \text{ y } \alpha \text{ reales positivos}$$

Tuvimos curiosidad de conocer al menos el comportamiento en general de los valores de velocidad de los frentes de onda, cuando el orden de reacción crece. Si trabajamos con la forma canónica de la ecuación (8), observamos que al variar n lo que cambia es sólo la expresión de la autocatálisis. Analicemos dicha expresión.

Graficando las funciones $f(c)$ para FKPP=FK(2), FK=FK(3), FK(4) y FK(5) tenemos:

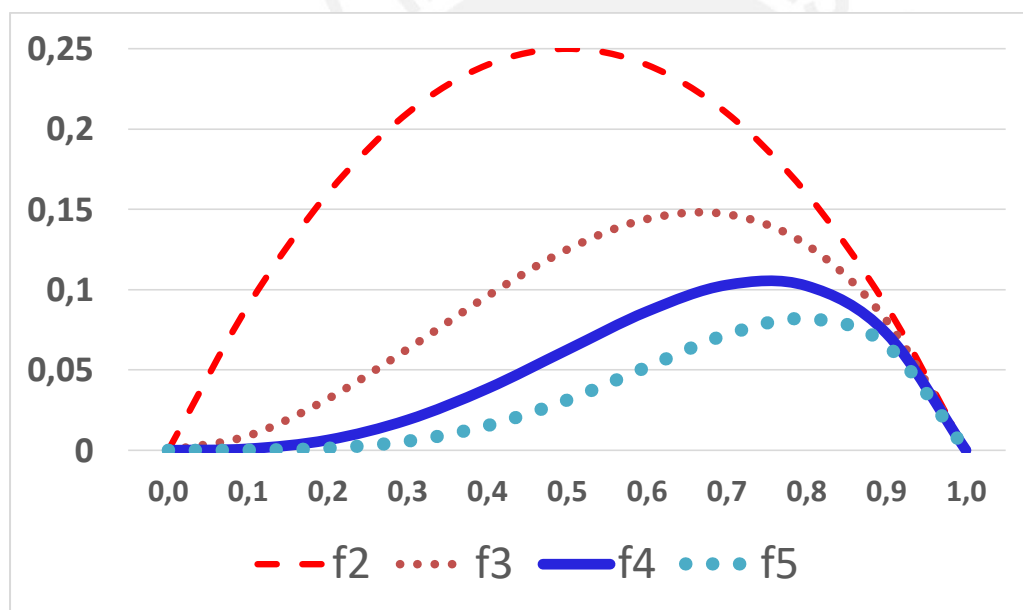


Figura 5. Gráfico de la función de reacción autocatalítica $f(c)$ para FK(2) a FK(5)

Lo que observamos en la figura 5 es que a nivel de funciones autocatalíticas, en el intervalo $0 < c < 1$, se cumple que:

$$c(1 - c) = f_2(c) > c^2(1 - c) = f_3(c) > c^3(1 - c) = f_4(c) > f_5(c) \dots \dots \dots (19)$$

En el **Anexo 2** demostramos que cuando $0 < c < 1$, $f_n(c) > f_{n+1}(c)$.

Dado que en la ecuación FK(n+1) lo que varía al cambiar n es la función autocatalítica, cuyo valor se hace cada vez menor, simplificando la expresión de la ecuación FK(n+1):

$$\frac{\partial c}{\partial t} = \dots + c^n(1 - c), \text{ con } n \geq 3,$$

Entonces

$$\frac{\partial c}{\partial t} = \dots + c^n(1 - c), \text{ **disminuye**, a medida que } n \text{ **aumenta** (20)}$$

Como la velocidad base $v_{(n+1)}$ asociada a la ecuación FK(n+1) está influenciada directamente por el comportamiento del término $\frac{\partial c}{\partial t}$, entonces podemos intuir razonablemente que:

$$\text{Velocidad base } v_{(n+1)} \text{ **disminuye** a medida que } n \text{ **aumenta** (21)}$$

Verificando la expresión (21) para FK(n+1) en forma canónica, observamos que las expresiones (16) y (17) cumplen con la antedicha condición:

$$2 = v_2 \geq v_3 = \frac{1}{\sqrt{2}}$$

$$\text{A partir de lo cual podemos intuir que } v_2 \geq v_3 \geq v_4 \geq v_5, \text{(22)}$$

Acto seguido, procedimos a simular las velocidades de los frentes en 1+1 dimensiones, aproximando la solución de las ecuaciones FK(4) y FK(5), cuyos resultados fueron:

Tabla 7: Velocidades base simuladas, ecuaciones FK(4) y FK(5), 1+1 dimensiones, órdenes 4 y 5

Orden de Reacción AutoCatalítica	Velocidad Base Simulada (V_{ns})
Cuatro	$V_{4s} = 0.00207125$
Cinco	$V_{5s} = 0.00154991$

Observamos que los resultados presentados en la tabla 7, sobre v_{4s} y v_{5s} cumplen la relación (22).

Conforme mencionamos en la introducción de este trabajo, luego de haber aproximado razonablemente los valores base para las velocidades de los frentes de onda, es decir, luego de haber aproximado los valores V_{4s} y V_{5s} , procedimos a estudiar el impacto del aumento de la velocidad transversal promedio \bar{U} sobre la velocidad final del frente de

onda, para reacciones autocatalíticas de orden cuatro y cinco. Recordemos que no contamos con una expresión teórica similar a la expresión (7), que nos servía para calcular la velocidad teórica en la ecuación $FK = FK(3)$.

En otras palabras, conforme explicamos en la introducción, en la fase 4 del presente trabajo, sólo era posible trabajar con dos expresiones para cada orden de reacción autocatalítica:

- Resolver la Ecuación $FK(4)$ y $FK(5)$ en 2+1 dimensiones, asumiendo un flujo transversal dado por Chatwin [12].
- Aproximar la solución de las ecuaciones $FK(4)$ y $FK(5)$, en 1+1 dimensiones, asumiendo que la expresión de la difusión efectiva $D_{eff} = D + D_T$ se mantiene, con

$$D_T = \frac{\bar{U}^2 b^2}{210D}$$

Para fines de comparación, luego de aproximar las velocidades finales, tanto para $FK(4)$ como para $FK(5)$, para cada valor del promedio de velocidad transversal \bar{U} , simulamos las velocidades finales obtenidas en la forma de velocidades normalizadas, para facilitar las comparaciones. Los resultados los mostramos en forma gráfica en las figuras 6 y 7.

En las antedichas figuras, podemos observar que para los órdenes de reacción 4 y 5 las velocidades normalizadas aproximadas mediante la difusión efectiva (en 1+1 dimensiones) y las velocidades simuladas usando 2+1 dimensiones, coinciden en mayor medida respecto de las reacciones cúbicas – figura 4. Dicho de otro modo, se observa que a mayor orden de reacción, es menor la diferencia entre la expresión asintótica (1+1 dimensiones) y la simulación con 2+1 dimensiones.

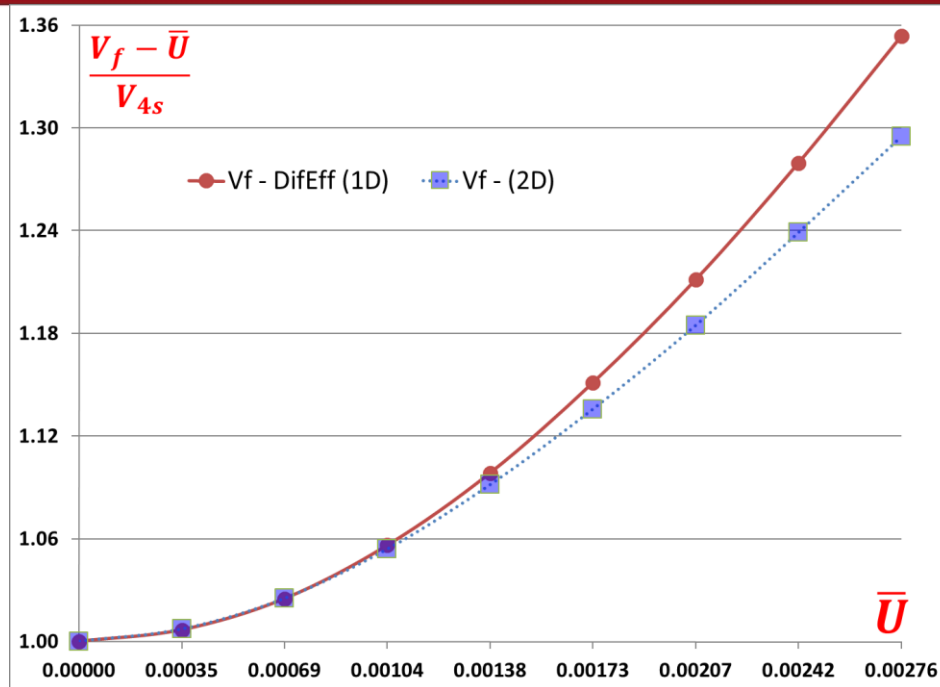


Figura 6. Velocidades normalizadas simuladas, reacción orden 4

Donde:

- V_f : velocidad final del frente
 V_{4s} : velocidad aproximada base para reacción autocatalítica
 \bar{U} : velocidad promedio, flujo transversal Chatwin [12]
 $V_f - \text{DifEff}(1D)$: velocidad final efectiva, aproximada en 1+1 dimensiones
 $V_f - (2D)$: velocidad final efectiva, aproximada en 2+1 dimensiones

Para el caso de la reacción autocatalítica de orden cinco, tenemos:

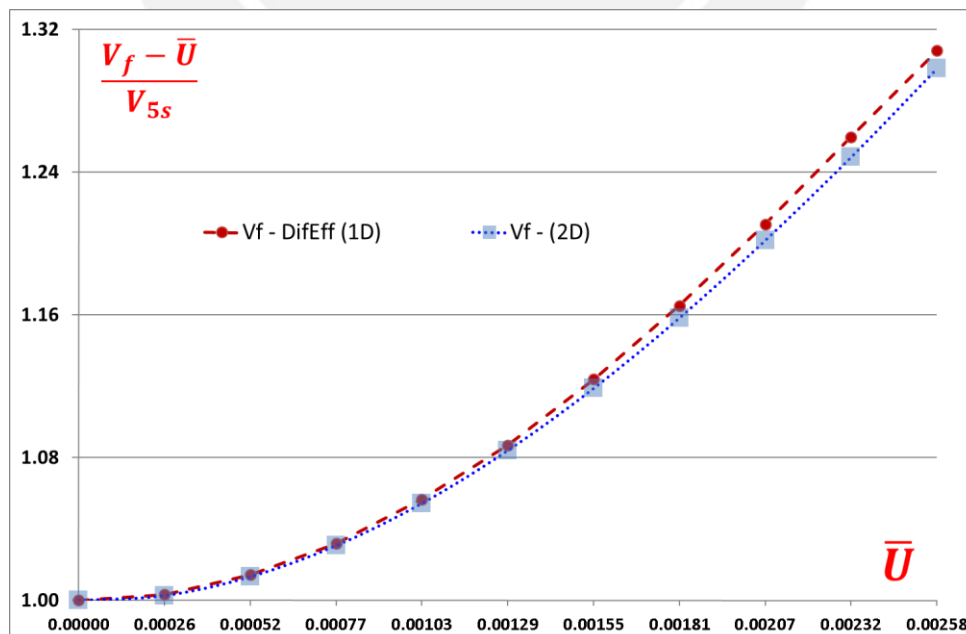


Figura 7. Velocidades normalizadas simuladas, reacción orden 5

Donde:

V_f :	velocidad final del frente
V_{4s} :	velocidad aproximada base para reacción autocatalítica
\bar{U} :	velocidad promedio, flujo transversal Chatwin [12]
$V_f - D_{\text{eff}} E_{\text{ff}}(1D)$:	velocidad final efectiva, aproximada en 1+1 dimensiones
$V_f - (2D)$:	velocidad final efectiva, aproximada en 2+1 dimensiones



9. Conclusiones

En el presente trabajo estudiamos y aproximamos las velocidades de los frentes de onda de reacciones autocatalíticas, siguiendo la estrategia de fases delineada en la introducción del presente trabajo.

Lo relevante al desarrollar nuestro trabajo fue:

- a. Se ha observado la aplicabilidad del mecanismo de difusión efectiva, para aproximar la velocidad de frentes de onda en 2+1 dimensiones, para órdenes de reacción cuatro y cinco, observándose mayor coincidencia en los valores de las velocidades finales estimadas, en tanto mayor sea el orden de la reacción autocatalítica.
- b. Hemos observado que los mecanismos de aproximación o cálculo de la velocidad final de los frentes de onda, usando la ecuación ADR (advección-difusión-reacción), en general son de más ayuda cuanto menor sea la magnitud de la velocidad máxima del frente transversal.
- c. Se ha comprobado la hipótesis que, al aumentar el orden de la reacción autocatalítica para la ecuación extendida de Fisher $FK(n+1)$ en 1+1 dimensiones, sin flujo externo, disminuye el valor de la velocidad de los frentes asociados.
- d. En el cálculo de las velocidades finales de los frentes de onda, ha sido de gran ayuda el criterio enunciado por U. Ebert and W. van Saarloos [14], mediante el cual obtenemos velocidades homogéneas de los frentes de onda, para valores del tiempo suficientemente grandes.

Adicionalmente,

- e. Durante el desarrollo del trabajo, se comprobó la importancia de seguir los criterios de convergencia y estabilidad al momento de seleccionar las longitudes mínimas de la grilla $\Delta t, \Delta x$ e Δy , para obtener valores consistentes de las concentraciones normalizadas y por consiguiente de las velocidades de los frentes de onda.

- f. El método FTCS nos ha sido de utilidad para la aproximación de las ecuaciones FKPP, FK y FK(n+1), tanto para 1+1 como para 2+1 dimensiones.
- g. Se ha validado la aplicabilidad de la grilla móvil, al replicar los resultados teóricos y simulaciones ya realizadas para reacciones autocatalíticas de orden 2 y 3.
- h. La grilla móvil es una herramienta que permite procesar formulas del tipo explícito, no recursivo, extendiendo de forma ordenada y modular la capacidad de procesamiento del equipo de cómputo.
- i. La grilla móvil es una herramienta flexible y efectiva, que nos permite adaptarnos a las características de hardware y software de la computadora donde se ejecuta la simulación, para aprovechar las facilidades de memoria cache L1-L2-L3.
- j. Los detalles de la implementación en Fortran de la grilla móvil han permitido reducir el tiempo de las simulaciones, al aprovechar los recursos de sistema disponibles: dimensión primaria el tiempo, uso de instrucciones **forall**, uso de **do** implícitos para operaciones de I/O, uso de parámetros para legibilidad y flexibilidad de uso de la implementación.

10. Trabajos Futuros

A partir de lo desarrollado en el presente trabajo, sugerimos que la grilla móvil implementada se utilice que investigar y/o simular lo siguiente:

- 10.1. Siguiendo el trabajo de Leconte et al. [10], calcular el impacto sobre la velocidad del frente de onda cuando variamos el ancho de la dimensión, es decir, investigar el efecto de variar lentamente el parámetro adimensional $\Phi = \frac{b}{l_x}$.
- 10.2. Calcular la velocidad base para órdenes de reacción mayor a 5, en 1+1 dimensiones, para tratar de encontrar una heurística de cálculo para las velocidades base con la función $f(c) = c^n (1 - c)$.
- 10.3. Simular la ecuación de Fisher – Kolmogorov, para 3+1 dimensiones.
- 10.4. Modificar el mecanismo de la grilla móvil para resolver ecuaciones con regiones de rápida variación (*stiff equations*), considerando particiones que adapten su longitud en dichas regiones.

11. Referencias

- [1] K. Showalter, "Quadratic and Cubic Reaction-Diffusion Fronts", Nonlinear Science Today, 1 Vol.4, Number 4 (1995)
- [2] G. Castellan, "Physical Chemistry", 3rd Edition, Addison Wesley Publishing Company Inc. (1983)
- [3] F. White, "Fluid Mechanics", 6th Edition, McGraw-Hill Series in Mechanical Engineering (2006).
- [4] M. Leconte, J. Martin, N. Rakatomalala, and D. Salin, "Mixing and reaction fronts in laminar flows", Journal of Chemical Physics, volume 120, number 16 (2004)
- [5] J. Hoffman, "Numerical Methods for Engineers and Scientists", Marcel Dekker Inc. New York, 2001.
- [6] J. D. Logan, "An Introduction to nonlinear Partial Differential Equations", 2nd Edition, Wiley-Interscience, a John Wiley & Sons Inc. Publication (2008)
- [7] R. L. Burden, J. L. Faires, "Numerical Analysis", 5th Edition, Cengage Learning (2011)
- [8] W. Cheney, D. Kincaid, "Numerical Mathematics and Computing", 6th Edition, Cengage Learning (2007)
- [9] A. R. Mitchell, D. F. Griffiths, "The Finite Difference Method in Partial Differential Equations", Wiley-Interscience, a John Wiley & Sons Inc. Publication (1994)
- [10] M. Leconte, N. Jarridge, J. Martin, N. Rakotomalala, D. Salin and L. Talon, "Taylor's regime of an autocatalytic reaction front in a pulsative periodic flow", Physics of Fluids 20, 057102(2008).

- [11] W. Hereman and W. Malfliet, "The Tanh Method: A Tool to Solve Nonlinear Partial Differential Equations with Symbolic Software", Journal of Computational and Applied Mathematics 03/2004; 164-165(1):529-541.
- [12] P. C. Chatwin, "On the longitudinal dispersion of passive contaminant in oscillatory flows in tubes," J. Fluid. Mech. 71, 513 (1975).
- [13] G. W. Griffiths, W. E. Schiesser, "Traveling Wave Analysis of Partial Differential Equations", Elsevier Inc., (2012).
- [14] U. Ebert and W. van Saarloos, "Front propagation into unstable states: Universal algebraic convergence towards uniformly translating pulled fronts," Physica D 146, 1 (2000)
- [15] J. C. Adams, W. S. Brainerd, J. T. Martin, B. T. Smith & J. L. Wagener, "Fortran 90 Handbook, Complete ANSI/ISO Reference", Intertext Publications Mc Graw Hill Book Company (1992)
- [16] L. R. Nyhoff, S. Leestma, "Introduction to Fortran 90 for Engineers and Scientists", Prentice Hall Inc. (1995)
- [17] D. A. Patterson, J. L. Hennessy, "Computer Organization and Design: the hardware/software interface", fifth edition, Morgan Kauffmann, Elsevier (2014)
- [18] A.S. Tanenbaum, T. Austin, "Structured Computer Organization", Pearson, Prentice Hall (2012)
- [19] http://ark.intel.com/es/products/75117/Intel-Core-i7-4700MQ-Processor-6M-Cache-up-to-3_40-GHz, accessed May 17th 2015.
- [20] D.R. Willé, "Advanced Scientific Fortran", John Wiley and Sons (1995).
- [21] A. Markus, "Modern Fortran in Practice", Cambridge University Press (2012)

- [22] R. Chandra, L. Dagum, "Parallel Programming in Open MP", Morgan Kauffmann Publishers (2001)
- [23] M. Hermanns, "Parallel Programming in Fortran 95 using Open MP", Universidad Politécnica de Madrid, Spain (2002)
- [24] O. Cornejo-Pérez, H.C. Rosu, "Nonlinear second order ODE: Factorizations and particular solutions", Prog. Theor. Phys. 114 (3) (2005) 533-538.
- [25] J. -H. He, X. -H. Wu, "Exp-function method for nonlinear wave equations", Chaos, Solitons and Fractals, 30 (2006) 700-708
- [26] D. Baldwin, U. Goktas, W. Hereman, L. Hong, R.S. Martino, J.C. Miller, "Symbolic computation of exact solutions expressible in hyperbolic and elliptic functions for nonlinear PDEs", Journal Symbolic Computation 37 (2004) 669-705
- [27] J. Hoste, "Mathematica Demystified", Mc Graw Hill (2009)
- [28] H. C. Rosu, O. Cornejo-Pérez, "Supersymmetric pairing of kinks for polynomial nonlinearities, Physical Review E 71, 046607 (2005)
- [29] M. J. Ablowitz, A. Zepetella, "Explicit solutions of Fisher's Equation for a special wave speed", Bulletin of Mathematical Biology, November 1979, Volume 41, Issue 6, pp 835-840
- [30] G. I. Taylor, "Dispersion of soluble matter in solvent flowing slowly through a tube," Proc. R. Soc. London, Ser. A 219, 186 (1953); "Diffusion and mass transport in tubes," Proc. R. Soc. London, Ser. B 67, 857 (1954)
- [31] C. I. Alves Ventura dos Santos, "Sistemas de liberación controlada de fármacos", Tesis Doctoral Universidad de Alcalá y Universidad de Coimbra (2012)
- [32] J. Masere, D.A. Vásquez, B. F. Edwards, J. W. Wilder and K. Showalter, "Nonaxisymmetric and Axisymmetric Convection in Propagating Reaction-Diffusion Fronts", J. Phys. Chem. 1994, 98, 6505-6508
- [33] D. A. Vasquez, "Chemical Instability Induced by a Shear Flow", Phys. Rev. Letters, volume 93, number 10, Sep. 2004
- [34] A. M. Turing, Philos. Trans. R. Soc. London B **327**, 37 (1952)

- [35] V. Cassets, E. Dulos, J. Boissonade and P. De Kepper, Phys. Rev. Lett. **64**, 2953 (1990)
- [36] B. F. Edwards, Phys. Rev. Lett. 89, 104501 (2002)
- [37] M. Leconte, J. Martin, N. Rakotomalala, and D. Salin, Phys. Rev. Lett. **90**, 128302 (2003)
- [38] A.D. Polyanin, V.F. Zaitsev, "Handbook of Nonlinear Partial Differential Equations", CRC Press (2012)

